LA-UR-03-1563

Introduction to Monte Carlo Methods: TITANS

Theoretical Institute for Thermonuclear and Nuclear Studies
March 10, 2003

John S. Hendricks
D-5 Design Safety and Risk Analysis
P.O. Box 1663, Mail Stop K575
Los Alamos, New Mexico 87545

Outline:

1. INTRODUCTION

1.1. Definition/History/Uses

This presentation can be summed up in one sentence: *The Monte Carlo method is a powerful statistical technique for modeling nature.*

The Monte Carlo method can be defined loosely as any mathematical technique that makes use of random numbers. The name comes from the resort town and site of the casino in Monaco. Such techniques are also called stochastic techniques.

The Monte Carlo method generally is attributed to scientists working on the development of nuclear weapons in Los Alamos during the 1940s. However, its roots go back much further.

Perhaps the earliest documented use of random sampling to solve a mathematical problem was that of Compte de Buffon in 1772. Laplace suggested in 1786 that π could be evaluated from random sampling. A century later, people performed experiments in which they threw a needle in a haphazard manner onto a board ruled with parallel straight lines and inferred the value of π from observations of the number of intersections between the needle and the lines. Lord Kelvin appears to have used random sampling to aid in evaluating some time integrals of the kinetic energy that appear in the kinetic theory of gases and acknowledged his secretary for performing calculations for more than 5000 collisions.

Enrico Fermi invented a form of the Monte Carlo method when he was studying the moderation of neutrons in Rome. Although Fermi did not publish anything, he amazed his colleagues with his predictions of experimental results. Only later would he reveal that his "guesses" were really derived from the statistical sampling techniques that he performed in his head when he could not fall asleep.

During World War II, Fermi joined Stan Ulam, John von Neumann, Robert Richtmyer, and Nicholas Metropolis in Los Alamos. These were the founders of the modern Monte Carlo method. They realized that statistical sampling techniques were impractical because they were long and tedious, but with the development of computers, they could become practical. Statistical sampling techniques reminded everyone of games of chance, where randomness statistically would become resolved in predictable probabilities. It was Nicholas Metropolis who noted that Stan Ulam's uncle would borrow money because he "just had to go to Monte Carlo" and thus named the mathematical method "Monte Carlo."

Since then the Monte Carlo method has become the method of choice for solving radiation transport problems. These problems model how radiation interacts with matter. Monte Carlo also is used for probabilistic risk analysis, solid state physics, equations of state, economics modeling, political and sociological modeling, and many other uses that will not be described further. Nonlinear radiation transport Monte Carlo, where the radiation changes the medium with which it is interacting, also will not be covered. This presentation is limited to radiation transport linear Monte Carlo.

1.2. Monte Carlo Methods vs Deterministic Methods

Nature is modeled in two ways: using deterministic methods and using stochastic methods. Deterministic methods involve modeling something with an equation which is then solved, or determined. The deterministic methods can be solved exactly, but the equations are usually approximate. On the other hand, the Monte Carlo method is as exact as we want it to be, but the solution is statistically approximate. Deterministic methods are global; the answer is solved everywhere at once, but it is an approximate

answer. Monte Carlo methods are local; any given point in phase space can be modeled in as much detail as desired, but then a final answer can never quite fully converge.

In brief, deterministic methods get the right answer to the wrong problem. Monte Carlo methods get the wrong answer to the right problem. Only in the limit of infinitely small deterministic space, energy, time, and angle meshes or in the limit of an infinite number of Monte Carlo statistical samples can both methods give the correct answer.

Faster computers and better Monte Carlo methods greatly have reduced the problem of slow statistical convergence for the Monte Carlo method. Because the Monte Carlo method is local, we can model what is happening in as much detail as we want. In the case of neutron and photon transport, where the physics is well understood, we can model what is happening nearly perfectly. This modeling results in a truly predictive capability for how radiation interacts with matter.

Furthermore, the Monte Carlo method is eminently realistic and very intuitive. In deterministic methods, the physics is hidden in mathematical arrays, and the method is cast in terms of abstract equations. In Monte Carlo, the focus is on the simulation of the actual physics that occur. To model Compton scattering with a deterministic method, the Boltzmann transport equation is solved, which bears little resemblance to the Klein-Nishina equation; in the Monte Carlo model, the Klein-Nishina equation is actually used.

2. FUNDAMENTALS

2.1. Cross Sections

To describe how the Monte Carlo method works for radiation transport problems, it is important to first consider the principal parameter of probability in such calculations: the cross section. The cross section, σ, is proportional to the probability that an incident particle will interact with a nucleus, atom, or other target. The units for cross sections are those of area because associated with every scattering center is an area σ such that an incident particle heading for it will interact. The units are "barns," or 10^{-24} cm^2. The name came about because early researchers thought the probability of some interactions was so great that it was like striking the "broad side of a barn."

The probability of an interaction is also proportional to the density of the target atoms. In a vacuum (zero density), there will be no interactions. In a high-density medium, there will be many interactions. The product of the atom density, ρ, and the cross section, σ, is the "macroscopic" cross section, $\Sigma = \rho\sigma$, with units of inverse centimeters.

2.2. Direct Sampling

Cross sections and probabilities are used in Monte Carlo sampling schemes. These schemes use random numbers to sample various physics phenomena statistically. Either direct or rejection sampling techniques can be used.

A direct sampling technique may be illustrated by sampling the distance to collision. The probability of a particle traveling a distance x and then colliding within dx is

$$p(x)dx = \rho\sigma_T e^{-\rho\sigma_T x} dx \quad .$$

Here, ρ is the atom density and σ_T is the total cross section. The fundamental principle for continuous probability density functions, *p(x)*, which is normalized as

$$\int_0^\infty p(x)dx = 1$$

is

$$\xi = P(x) = \int_0^x p(t)dt \quad .$$

This equation determines *x* uniquely as a function of ξ , where ξ is a random number. Moreover, if ξ is distributed uniformly on $0 < \xi < 1$, then *x* falls with frequency *p(x)dx* in the interval (*x*, *x* + *dx)*. Thus, to sample directly the distance to a collision,

$$\xi = P(x) = \int_0^x \rho\sigma_T e^{-\rho\sigma_T t} dt = 1 - e^{-\rho\sigma_T x}$$

and

$$x = -\frac{1}{\rho\sigma_T} ln(1-\xi) \quad .$$

However, because $1 - \xi$ is distributed in the same manner as ξ, it may be replaced by ξ and we get

$$x = -\frac{1}{\rho\sigma_T} ln(\xi) \quad .$$

2.3. Random Numbers

Modern Monte Carlo calculations do not really use random numbers. If the numbers were truly random, then the results would change every time a calculation was made. Consequently, pseudo-random numbers are used. A mathematical algorithm, known as a random number generator, is used to come up with the same sequence of pseudo numbers for every calculation. These random numbers are chosen so that they have random characteristics, such as the proper mean and distribution. Generally, the *nth* random number is the fraction

$$0 < \xi_n < 1 \quad .$$

Random number generators can be relatively simple. Our production codes use the "congruential method" invented in the 1950s.

$$\xi_{n+1} = A\xi_n \quad .$$

If the old random number, ξ_n, the multiplier, A, and the new random number, ξ_{n+1} are all 64-bit numbers on a computer, then we use only the 64 least significant bits of the 128-bit product, $A\xi_n$, to generate the new random number ξ_{n+1}. Mathematicians get all excited about new and better random number

generators, but having a better random number generator is seldom consequential in radiation transport problems.

2.4. Error Estimation: The Central Limit Theorem

A statistical estimation of anything is of little value without knowing the statistical error. The Central Limit Theorem of statistics enables errors to be estimated.

The Monte Carlo method may be thought of as the solution to an integral:

$$\int f(x)dx = 1 \quad ,$$

where $f(x)$ is the underlying probability density function for a given problem normalized so that the integral is unity. Seldom is $f(x)$ as simple as the exponential function from which we sampled to get the distance to collision. Usually, the underlying probability density function is very complicated and unknown because it is a function of all the geometric, physical, and mathematical details of the entire problem.

The Monte Carlo tallies estimate the statistical "mean" value of the underlying problem probability distribution function

$$M_1 = \int xf(x)dx \quad ,$$

where M_1 is the first moment of the underlying probability distribution function and is also the "expected value." In the Monte Carlo simulation, the expected value, or tally mean, is estimated as the average value of scores, x_i averaged over all N histories of the Monte Carlo calculation:

$$t_1 = \frac{1}{N}\sum x_i \quad .$$

To estimate errors, the second moment is also used:

$$M_2 = \int x^2 f(x)dx \quad ,$$

which is estimated in the Monte Carlo calculation as

$$t_2 = \frac{1}{N}\sum x_i^2 \quad .$$

According to the Central Limit Theorem of the theory of probability, the distribution of the sum of N independent, identically random variables (Monte Carlo history estimators, x_i, in our case) with finite means and variances approaches a normal distribution as N takes on large values. Consequently, there is a standard deviation, σ, such that the tally estimate of the mean, t_1, is within a standard deviation, S, of the true mean, M_1:

$$\lim_{N\to\infty} P\{M_1 - aS \le t_1 \le M_1 + bS\} = \frac{1}{\sqrt{2\pi}}\int_a^b e^{-t^2/2}dt \quad .$$

That is, the probability that the estimated mean, t_1, as $N \to \infty$, is between M_1 - aS and M_1 + bS, can be calculated from a normal distribution, which is the expression on the right. If a = 1 and b = 1, then the expression on the right is approximately two-thirds and the estimated mean, t_1, is therefore within the bounds of the true mean, $M_1 \pm S$ two-thirds of the time. The standard deviation, S, is related to the variance as

$$S = \frac{\sigma}{\sqrt{N}} \quad .$$

The variance is given by

$$\sigma^2 = \int_{-\infty}^{\infty} (x - M_1)^2 f(x) dx$$

$$= \int_{-\infty}^{\infty} x^2 f(x) dx - 2M_1 \int_{-\infty}^{\infty} x f(x) dx + M_1^2 \int_{-\infty}^{\infty} f(x) dx$$

$$= M_2 - M_1^2 \quad .$$

The Monte Carlo estimate of the variance is then

$$\sigma^2 \approx t_2 - t_1^2 \quad .$$

In most Monte Carlo calculations, we are interested in the relative error, R, which is the standard deviation normalized by the tally mean, or

$$R = \frac{S}{t_1} = \frac{\sigma}{\sqrt{N t_1}} \quad .$$

The above equation shows why the Monte Carlo error estimate decreases as the square root of the number of histories, N. The relative error is therefore

$$R = \frac{\sigma}{t_1 \sqrt{N}} = \frac{\sqrt{t_2 - t_1^2}}{t_1 \sqrt{N}}$$

$$= \sqrt{\frac{t_2}{N t_1^2} - \frac{1}{N}} = \sqrt{\frac{\sum x_i^2}{(\sum x_i)^2} - \frac{1}{N}} \quad .$$

2.5. Rejection Sampling: π

The rejection sampling of π illustrates many of the fundamentals of Monte Carlo: the use of random numbers and sampling and the estimation of errors. Rejection sampling is an alternative to direct sampling. Usually both methods can be used; the sampling efficiency determines which method is chosen.

Suppose we wish to calculate the value of π by Monte Carlo. We inscribe a circle of radius *r* into a box with side *2r*. The area of the circle is $A_c = \pi r^2$. The area of the box is $A_b = 4r^2$. Thus, $\pi = 4A_c/A_b$. To sample the value of π, we sample the area of the box uniformly and tally a score of 4 whenever the part of the box sampled is also in the circle; we tally a score of 0 when the part of the box sampled is not in the circle. If we have *N* samples, we will score 4 proportionately to the time we sample inside the circle, A_c/A_b, and thus estimate π.

Suppose we have 10 samples and 7 are sampled in the circle. Then the estimate of π is

$$t_1 = \frac{1}{N}\sum x_i$$

$$= \frac{1}{10} 7 \times 4 = 2.8 \quad .$$

The second moment is

$$t_2 = \frac{1}{N}\sum x_i^2$$

$$= \frac{1}{10} 7 \times 16 = 11.2 \quad .$$

The relative error is

$$R = \sqrt{\frac{t_2}{N t_1^2} - \frac{1}{N}} = \sqrt{\frac{\sum x_i^2}{(\sum x_i)^2} - \frac{1}{N}}$$

$$= \sqrt{\frac{11.2}{78.4} - \frac{1}{10}} = \sqrt{.04286} = 0.207 \quad .$$

Thus, the estimate of π is

$2.8 \pm .6$,

or

$2.2 < \pi < 3.4$.

It is evident that many more samples, or histories, must be run to get a more accurate result. The following table shows the result of a short program to sample π. The number of particles started is nps and the hits are the number of times the rejection sampling hit in the circle.

Output from Short Monte Carlo Program to Sample π

nps	hits	mean	rel err	true err
10.	8.	3.2000	0.1581	0.0186

100.	77.	3.0800	0.0547	0.0196
1000.	785.	3.1400	0.0165	0.0005
10000.	7874.	3.1496	0.0052	0.0025
100000.	78545.	3.1418	0.0017	0.0001
200000.	157203.	3.1441	0.0012	0.0008
300000.	235634.	3.1418	0.0010	0.0001
400000.	314097.	3.1410	0.0008	0.0002
500000.	392884.	3.1431	0.0007	0.0005
600000.	471319.	3.1421	0.0007	0.0002
700000.	549761.	3.1414	0.0006	0.0001
800000.	628385.	3.1419	0.0006	0.0001
900000.	706851.	3.1416	0.0006	0.0000
1000000.	785360.	3.1414	0.0005	0.0000

Finally, note again that either direct sampling or rejection sampling can be used for most functions. In direct sampling we solve for the variables of density functions in terms of random numbers. Note that we also could sample the function $p(x) = \rho\sigma_T e^{-\rho\sigma_T x}$ by rejection by drawing the function inside a box from $0 < p(x) < 1$ high and $0 < x < \infty$ and then rejecting all samples above the curve and accepting all samples below the curve. However, the efficiency of this scheme is $1/\infty = 0$. Likewise, we could also sample π directly by sampling the height of the box uniformly and averaging the horizontal chord length within the circle. If the circle is of radius 1 and the sampled height is $y = \xi$, then the horizontal chord length in the quadrant is

$$x = \sqrt{1-\xi^2} \quad .$$

3. THE RANDOM WALK

3.1. Shielding

To illustrate how the Monte Carlo method works and how it is eminently realistic, consider a typical radiation transport shielding problem. The goal is to get particles (radiation) from a source, through a geometry, to a tally region or some other place. For example, suppose we are designing the shielding for the Dual Axis Radiographic Hydrodynamics Test (DARHT) Facility. We have a source of radiation on one side of the wall, and we want to know what gets out on the other side. We simply solve the problem by having a mathematical simulation of the physical situation. We follow one mathematical particle after another as it bounces through the wall. Some particles get absorbed. Some particles bounce back out the source wall. Some particles penetrate. This occurs in nature, and this occurs in the Monte Carlo simulation. Statistical laws govern what happens in nature, and statistical laws govern what happens in the Monte Carlo calculation.

First, we model the source. If the true physical source is a plane wave, we start our Monte Carlo particles in a direction normal to the surface directed inward. If the true physical source is isotropic, we start with a cosine distribution. If the true physical source is anything else, if it can be characterized, we can "sample" it and model it exactly as it is characterized. If the source is distributed in some way over the face of the wall, we can model any known distribution. If the source particles have an energy distribution, we can sample from that distribution. Sampling methods involve random numbers to pick variables from a distribution, much like throwing dice or gambling at the Monte Carlo casino. The point is that because we have to model only one source particle at a time at one particular location in phase space, we can model it in as much detail as we like.

Thus, the Monte Carlo simulation begins by throwing the dice to sample the initial phase space variables of the starting source particle history. Once a source particle is modeled, we know its location, x, y, z; its direction, with direction cosines u, v, w; its energy, E; and its time, T. We also need to know its weight, W, which is the physical number of particles represented by the statistically sampled particle. Usually, Monte Carlo calculations are normalized to 1 source particle, so the starting weight is 1.

We now do the "random walk." We statistically sample what happens to the particle as it traverses the geometry: in this case, the DARHT wall. We use random numbers, or throw the dice, to choose between events such as capture, scatter, (n,xn), or fission, according to their probabilities, or cross sections. Thus, the random walk is just a matter of deciding what the probabilities of various events are and then using random numbers, or throwing the dice, to choose between them.

3.2. Criticality

Another illustration of how the Monte Carlo method works is the calculation of nuclear criticality. Nuclear criticality is an eigenvalue problem in which we estimate the value of k_{eff}, which is the number of neutrons produced from fission in one generation per number of neutrons produced in the previous generation. If $k_{eff} < 1$, then the system being modeled is subcritical. If $k_{eff} = 1$, then the system is critical and we have a self-sustaining nuclear chain reaction. If $k_{eff} > 1$, then the system is supercritical and we have a nuclear accident or explosion.

The Monte Carlo random walk works by running the equal numbers of histories in batches, or cycles, representing each generation of neutrons. We start with source neutrons distributed in volume throughout all fissionable materials in the problem. Each neutron is isotropically emitted with its energy from a fission spectrum. These neutrons then random walk through the geometry until they escape, are absorbed, or undergo fission. Whenever a neutron has a fission, the fission neutrons are not followed further. Rather, they are stored as source neutrons for the next generation, or cycle.

At the end of each cycle, k_{eff} is estimated as the ratio of the number of fission neutrons created per source neutron in that cycle. Then the newly created fission neutrons are eliminated randomly or duplicated so that the number of source neutrons is the same for each cycle. The number of fission neutrons created per source neutron in each cycle is

$$k_{eff} = \rho \int \phi \nu \sigma_f dV \quad ,$$

where ρ is the atom density, ϕ is the flux, ν is the number of neutrons emitted in a fission, σ_f is the fission cross section, and V is the volume.

3.3. Estimators

The result of the random walk is that the particles we are following in our model do just what the physical particles do. If we are interested in how many traverse the wall, then we simply count them if and when they cross the outer-wall boundary. This counting is known as a "tally." We do tallies with estimators. A surface estimator merely counts particles as they cross a surface. Other tallies can be more elaborate.
The four Monte Carlo estimators are surface, event, track length, and next event.

3.3.1. Surface Estimators

Surface estimators are relatively simple. Scores are made when particles cross geometric surfaces. Typically, the weight of the particles is scored, which is in nuclear engineering jargon the "surface

current." The surface estimator is used in the homework problem to determine how many particles cross the slab wall.

3.3.2. Event Estimators

Collision estimators are the most common event estimators. Event estimators are estimators that are made at each event. The event estimator was used in both the calculation of π and the distance to collision examples described earlier. Collision estimators also are used to calculate reactions rates, such as the fission rate, in which case the weight times cross section is usually scored.

3.3.3. Track Length Estimators

Track length estimators are used to calculate quantities over volumes in space, such as particle flux. In nuclear engineering jargon, the particle flux is the particle density times velocity:

$$\phi = \eta v \quad .$$

In Monte Carlo, the time-integrated flux, or fluence, is then

$$\int \phi \, dt = \eta v dt = \frac{W}{V}\frac{dl}{dt}dt = W\frac{dl}{V} \quad ,$$

where *dl* is the track length between events (source, collision, and surface crossing); W is the particle weight; and V is the volume. Thus, the flux is estimated by the track length per unit volume.

In the limit of an infinitely thin volume, $V \to Adx$, where *dx* is the thickness of the volume and *A* is the surface area. The track length is $dl \to dx/\mu$, where μ is the cosine between the surface normal and the particle trajectory. Thus, the surface estimator of the flux on a surface is related to the track length estimator of the flux in a volume as

$$W\frac{dl}{V} \to \frac{W}{A\mu} \quad .$$

Reaction rates over a volume commonly are estimated by track length estimators. The total heating by particles passing through a volume is estimated as

$$\frac{p_a}{p_g}\int \phi\sigma_T H dt = \frac{p_a}{p_g}W\frac{dl}{V}\sigma_T H \quad ,$$

where p_α and p_g are the atom and gram densities, σ_T *is* the total cross section, and *H* is a heating number in millielectron volts per gram.

3.3.4. Next-Event Estimators

Next-event estimators are the most complicated estimators. Suppose we want to determine the flux at a point. The probability of having a random walk of Monte Carlo particles exactly to a point is zero. Even if the point is finite in size, it may take billions of statistical trials, or histories, to find a way there randomly.

The next-event estimator does not require a random walk particle to get to the point. Instead, at each event, the random walk proceeds as usual, but an estimate is made *as if* the next event were to go straight to the point. The estimate of the flux at the point for the next event is

$$\phi = \frac{We^{-\rho\sigma_T x} 2p(\mu)}{4\pi R^2} \quad ,$$

where $e^{-\rho\sigma_T x}$ is the attenuation through all materials between the event and the next event; $2p(\mu)$ is the probability density function for scattering toward the point where the flux is detected, with μ being the cosine of the angle between the incident particle trajectory and the flight path to the point; $4\pi R^2$ is the solid angle attenuation; and R is the distance from the event to the point of the next event.

With the next-event estimator, improbable events are now very probable. Instead of making an estimate once every zillion histories when a particle random walks to the point, an estimate is made for every source or collision event of every history.

4. VARIANCE REDUCTION

Because Monte Carlo calculations take a long time to converge to an acceptable answer, they were once considered to be the "method of last choice." However, modern computers have sped up Monte Carlo calculations by factors of thousands in the past few decades. Variance reduction methods have sped up Monte Carlo calculations by factors of millions.

A variance reduction technique reduces the variance of a Monte Carlo calculation for a fixed problem running time. As an alternative, the variance can be held constant, and the variance reduction technique reduces the running time. For example, suppose an analog calculation, one that directly simulates the physical problem without variance reduction, takes 1 hour to calculate a tally with a relative error of 10%. To reduce this error to 5%, we could (1) run the problem for four times as many histories, which would take 4 hours (because the relative error decreases as the square root of the number of histories); (2) use variance reduction to reduce the error to 0.00001%; or (3) use variance reduction to achieve a 5% variance in 1 second. Variance reduction techniques are extremely powerful.

The four classes of variance reduction methods are problem truncation, population control, modified sampling, and pseudo-deterministic methods.

4.1. Problem Truncation

Problem truncation is used in every Monte Carlo problem. We simply choose to ignore parts of the problem. For example, in modeling the physics of a room, we do not continue the model to infinity by modeling the Eiffel Tower in Paris. We also do not model every nut and bolt in a machine. We do not model energies or time in ranges where they cannot affect the problem outcome.

Although problem truncation is the simplest variance reduction technique, it is also the most dangerous to use. Further, it is the only biased estimator in that it always results in underpredicting results because any contribution to the finally tally from parts of truncated phase space are omitted. Common truncation methods include geometry truncation, time cutoff, and energy cutoff.

4.2. Population Control

The most popular form of variance reduction is population control. At any time we wish, we can split the histories we are following into two or more histories, provided we adjust the particle weight accordingly. Therefore, for example, we can assign a geometric region an "importance" twice that of the source region. Whenever a particle of weight W enters the higher-importance region, it is split into two particles

of weight $W/2$. In this way, we have more statistical samples in regions important to our problem, but because the total weight is conserved, we will still converge to the correct answer.

Likewise, we can "Russian roulette" particles going from a higher-importance region into a lower-importance region. If the importance change between regions is a factor of two, then Russian roulette works by terminating, or killing, a particle going into the region with a probability of one-half, but giving those particles that survive time twice the weight. In this way, we have fewer statistical samples in regions unimportant to our problem, but because the total weight is conserved on average, we still have an unbiased estimator and converge to the correct answer.

Population control methods allow us to have any number of statistical samples we want in any part of our problem we want. Consequently, we have lots of statistical samples in important regions of the problem, which reduces the variance by having better sampling, and we have fewer statistical samples in unimportant regions, which reduces the problem time (thereby reducing the variance per unit time) by not wasting time on them.

Common population control methods include geometric splitting or "importance sampling," "weight windows," weight cutoff (in which particles dropping below a given weight play Russian roulette rather than problem truncation by termination), and energy and time splitting/roulette.

4.3. Modified Sampling

The third class of variance reduction methods is modified sampling. At any time we can sample from a different distribution than the analog, or true physical, distribution as long as we modify the particle weight by the ratio of the true sampling probability density function to the one from which we sampled.

Here are some examples. Sources may be biased in energy, time, or direction. Suppose you have a photon source with two equiprobable source lines: 100 keV and 10 MeV, each chosen with equal probability, 0.5. If the 10-MeV line contributes 90% of your tally result, you may wish to sample it 90% of the time (with probability 0.9). The weight adjustment would be the ratio of the true probability to the biased probability, 0.5/0.9 = 5/9. The expected starting weight would still be 90% x 5/9 = 1/2. Likewise, the 100-keV line would be sampled 10% of the time (with probability of 0.1), but the starting weight would be 0.5/0.1 = 5 so that the expected starting weight from this line would still be 10% x 5 = 1/2.

Another example of modified sampling is "path length stretching," in which the cross sections are modified from σ_T to σ'_T. If $\sigma'_T < \sigma_T$, then the distance to collision is increased from $x = -\frac{1}{\rho\sigma_T} ln(\xi)$ to $x = -\frac{1}{\rho\sigma'_T} ln(\xi)$: that is, by a factor of $\frac{\sigma_T}{\sigma'_T}$. However, the weight must be adjusted by the ratio of the probability density functions,

$$W' = W \frac{p(x)}{p'(x)}$$

$$= W \frac{\sigma_T e^{-\rho\sigma_T x}}{\sigma'_T e^{-\rho\sigma'_T x}}$$

$$= W \frac{\sigma_T}{\sigma'_T} e^{\rho(\sigma'_T - \sigma_T)x} \quad .$$

Path length stretching in a shielding problem will cause more particles to penetrate the shield and score, thus increasing problem efficiency and reducing the variance. Other examples of modified sampling are forced collisions, collision energy bias, collision angle bias, and reaction selection bias.

Thus, not only can we put particles anywhere we want to, but also we can sample from any convenient probability density function we desire, as long as we adjust the weight accordingly. The goal in all cases is to modify the Monte Carlo random walk so that we have more samples and spend more time in regions of phase space that contribute to our tallies and less time and fewer samples in regions of phase space that do not contribute to our samples.

4.4. Pseudo-Deterministic Methods

The final class of variance reduction techniques is the most complex: pseudo-deterministic methods. Hybrid Monte Carlo deterministic methods are an example in which neutron thermalization is treated with the deterministic diffusion equations rather than Monte Carlo, which wastes much time because particles just scatter back and forth without doing much of interest in the process of diffusing. Another example is the "DXTRAN" method, in which the next-event estimator is used to transport particles from events that occur in the random walk to a sphere somewhere else in phase space where the particles would have a difficult time going otherwise. Then the particles continue their random walk with inside the sphere with the weight lowered by the next-event estimate,

$$\frac{We^{-\rho\sigma_T x} 2p(\mu)}{4\pi R^2} \quad .$$

The DXTRAN technique is kept unbiased by killing any particles that actually get to the DXTRAN sphere by random walk so that on average, the total particle weight is preserved.

5. CONCLUSION

5.1. Radiation Transport Applications

The Monte Carlo method has a wide variety of applications for radiation transport. Experiments are expensive; calculations are cheap. Radiation is dangerous; calculations are safe. Some environments, such as outer space, can only be calculated.

The Monte Carlo method is used to design experiments and their detectors. Many detectors used for airport security are designed with Monte Carlo. This method has many defense science applications, such as vulnerability, hardness, lethality, and effects. Monte Carlo also has medical applications, such as cancer-therapy radiation-treatment planning and the design of accelerators for medical applications. The Monte Carlo method is used for reactor analysis, criticality safety, radiation shielding, fuel management, and reactor detector design. Another application is the assay of bulk materials, such as quality control for batches of cement or monitoring of effluents. Monte Carlo is used to design well-logging tools used in oil and gas exploration. Space applications include detector response analysis, shielding for astronauts, shielding for electrical components, and space nuclear power. Monte Carlo was essential for the analysis of data in the recent discovery of ice on Mars.

Nearly everything having to do with radiation, from food irradiation to waste treatment, from medicine to military applications, from underground well logging to outer space, has benefited from the Monte Carlo method.

5.2. Summary

The Monte Carlo method is a powerful statistical technique for modeling nature.

We have seen that there are two methods of modeling nature in computational science: deterministic methods and statistical methods such as the Monte Carlo method. The strengths of the Monte Carlo method are the following.

1. *Realistic:* the Monte Carlo method is eminently realistic and intuitive. The random walk simulates the probabilistic nature of the physical world. On the other hand, deterministic methods solve abstract equations. The physics is all buried in the databases.

2. *Detailed modeling*: Because the Monte Carlo method needs to know only about the point in phase space where it is at any given time, it can model nature in as much detail as desired. Therefore, it is physically accurate. Almost any physical distribution can be sampled, by rejection, directly, or both. Nearly any information we want from the Monte Carlo simulation can be extracted by four kinds of estimators. The deterministic methods, on the other hand, only approximate the physical problem and are much more limited in the information they can provide.

The weaknesses of the Monte Carlo method are the following.

1. *Local answers*: Methods are global and solve the problem everywhere. Monte Carlo methods are local, and by focusing on "important" regions of phase space, information at other regions is lost. Thus, Monte Carlo methods are not as efficient for problems where an answer is desired everywhere.

2. *Statistical answers*: Deterministic methods exactly solve the discretized transport equation. Monte Carlo methods only converge statistically toward the correct solution. However, the Monte Carlo relative error can be readily calculated, making it possible to estimate how close the Monte Carlo estimators are to the true solution.

3. *Slow convergence*: The Monte Carlo method converges only as $1/\sqrt{N}$. However, variance reduction techniques have sped up the Monte Carlo method greatly; it is now competitive with deterministic methods in two dimensions and faster in three.

We have shown enough of the fundamentals of the Monte Carlo method so that you can write your own Monte Carlo computer code. However, do not be deceived into thinking production Monte Carlo codes are simple. Writing a general-purpose radiation transport Monte Carlo code, with arbitrary three-dimensional geometries, time-dependence, fully detailed continuous-energy physics for multiple particle types over a broad energy range, with lattice geometry capabilities, eigenvalue calculators, multigroup options, general sources, flexible tallies, false convergence detection, etc., plus multitasking, graphics, documentation and quality control, would require $10 to $100 million to start from scratch. The Monte Carlo codes at Los Alamos and the databases behind them represent hundreds, even thousands, of person-years of development. These codes truly have become the repositories of a vast body of physics knowledge.

The future of the Monte Carlo method for computational radiation transport physics is very bright. It is one of two ways of solving computational physics radiation transport problems, and its weaknesses relative to the alternative method keep getting resolved while its strengths become stronger and more compelling. Monte Carlo is a crucial methodology for advancement of the mission of Los Alamos National Laboratory.